IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Inventors: Peter Joseph Giacomini et al.

Serial No.: 09/725737

Filing Date: 11/29/2000

Art Unit: 2142

Examiner: Thong H Vu

Docket No.: 500-002US

Title: Method and Apparatus For Economical Cache Population

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

APPEAL BRIEF UNDER 37 CFR 41.67

Pursuant to 37 CFR 41.67, this brief is filed in support of the appeal in this application.

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party of interest in this application is the assignee of this application: Broadspider Networks, Inc. The applicant's attorneys, Jason Paul DeMont and Wayne S. Breyer, are shareholders in Broadspider Networks, Inc.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

STATUS OF CLAIMS

Claims 1 through 32 stand rejected and are being appealed.

STATUS OF AMENDMENTS

All amendments have been entered. There have been no amendments made subsequent to the final rejection.

SUMMARY OF THE CLAIMED SUBJECT MATTER

The present invention relates to data processing systems and computer networks in general, and, more particularly, to techniques for storing resources in a cache.

In the context of a computer network, when a user of the World Wide Web requests a Web page, the user must wait until the page is available on his or her data processing system (*e.g.*, computer, *etc.*) for viewing. In general, this wait occurs because the request for the Web page must traverse the Internet from the user's data processing system to the data processing system that is the source of the page, the request must be fulfilled, and the requested page must travel back to the user's system. If the Internet is congested or the data processing system that is the source of the page is overwhelmed, the wait can be considerably long. [Specification Page 1, Lines 10-16]

To shorten this wait, special data processing systems are deployed throughout the Internet to expedite the delivery of some Web pages. Some of these data processing systems expedite the delivery of Web pages by functioning as cache memories, which are also known as "caches." For example, a cache stores commonly requested Web pages and thereafter intercepts and fulfills requests for those pages, which eliminates the need for the request having to travel to, and be serviced by, the principal memory. [Specification Page 1, Lines 17-22]

A cache expedites the delivery of the Web page in two ways. First, a cache eliminates the need for the request to travel all of the way to the system that is the ultimate source of the page, and, therefore, eliminates some of the wait associated with the transit. Second, a cache also reduces the number of Web page requests that must be fulfilled by the system that is the ultimate source of the page, and, therefore, the wait associated with contention for the system is eliminated. [Specification Page 1, Lines 18-27]

Caches are also used to expedite processing in data processing systems and their operation in data processing systems is roughly analogous to, with notable exceptions, their operation in computer networks. At this time, the applicants do not believe that the differences between the use of cache in computer networks and in data processing systems are germane to the present appeal and the scope of the pending claims encompass the use of the present invention in both computer networks and data processing systems.

When a cache is full and an additional resource is to be stored in the cache, the cache must have an algorithm for deciding which resource gets purged to make room for the additional resource. There is a lot of prior art in this area.

In contrast, there is not a lot of prior art addressing the issue of when to store the resource in the cache in the first place. Typically, a resource is stored in a cache either (1) the first time that it is requested, or (2) it is pre-fetched before the first time that it is requested. The present invention addresses the issue of when to store a resource in a cache that is particularly efficient in many cases.

For example, a cache in accordance with the present invention is populated with a resource only when at least *i* requests for the resource have been received, wherein *i* is an integer and is, at least occasionally, greater than one. For example, a cache in accordance with the illustrative embodiment might not be populated with a resource unless two requests for the resource have been received within 10 minutes. In general, embodiments of the present invention are advantageous because *they prevent the cache from being populated with infrequently requested resources*. [Specification Page 3, Lines 21-31]

Here's a way to keep all of these ideas straight in one mind. There are only three ways that a cache can populated:

- **Pre-Filling** (AKA Pre-Fetching) — Populating a cache with a resource before a request for the resource has arrived.
- **Post-Filling** — Populating a cache with a resource after one request for the resource has arrived.
- **Delayed Post-Filling** — Populating a cache with a resource after more than one request for the resource has arrived.

Pre-filling and post-filling are old and well-known in the art. The present invention can be thought of as delayed post-filling and the combination of post-filling and delayed post-filling.

GROUNDS OF OBJECTION AND REJECTION TO BE REVIEWED ON APPEAL

Ground 1: 35 U.S.C. 112 Rejection of Claims 1, 8, 15, and 24

Claims 1, 8, 15, and 24 were rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the applicants regard as the invention.

Ground 2: 35 U.S.C. 102 Rejection of Claims 1-32

Claims 1 through 32 have been rejected under 35 U.S.C. 102(e) as being anticipated by S. Ali et al., U.S. Patent 5,896,506 (hereinafter "**Ali**").

Ground 3: 35 U.S.C. 102 Rejection of Claims 1-32

Claims 1 through 32 have been rejected under 35 U.S.C. 102(e) as being anticipated by V. Tran et al., U.S. Patent 7,039,683 B1 (hereinafter "**Tran**").

ARGUMENTS

Ground 1: 35 U.S.C. 112 Rejection of Claims 1, 8, 15, and 24

Claims 1, 8, 15, and 24 were rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the applicants regard as the invention. In particular, the Office action asserts that it is unclear How and When occasionally or under What condition the integer will be greater than one. The applicants respectfully traverse the rejection.

In order to comply with 35 U.S.C. 112, second paragraph, the claims do not need to specify how, when or under what conditions the integer *i* will be greater than one. Such a detail describes the invention. In contrast, 35 U.S.C. 112, second paragraph, requires that the applicant define the invention, and there is no vagueness or ambiguity in the scope of the claim as it currently reads.

The limitation at issue is this: wherein *i* is an integer and is at least occasionally greater than one. This is equal to reciting that there exists at least one instant in calendrical time *t* when $i > 1$, wherein *i* is an integer. Therefore, the applicants respectfully submit that the rejection is traversed.

If the utility of this limitation is in dispute, consider, for example and without limitation, a cache that has different criteria for caching resources based on the time of day. From Midnight until 7:00 AM, the cache is populated with a resource when one request for the resource has been received. From 7:00 AM until Midnight, the cache is not populated with a resource unless there have been two requests for the resource have been received. In other words, from 7:00 AM until Midnight, the cache is not populated with a resource until two requests for the resource has been received.

For these reasons, the applicants respectfully submit that the rejection is traversed.

Ground 2: 35 U.S.C. 102 Rejection of Claims 1-32

Claims 1 through 32 have been rejected under 35 U.S.C. 102(e) as being anticipated by S. Ali et al., U.S. Patent 5,896,506 (hereinafter "**Ali**").

Claim 1 recites:

> **1.** A method comprising:
>
> *populating a cache with a resource only when at least i requests for said resource have been received*;
>
> wherein *i* is an integer and is at least occasionally greater than one.
>
> (*emphasis supplied*)

Nowhere does Ali teach or suggest, alone or in combination with the other references, what claim 1 recites – namely, populating a cache with a resource only — in some situations — upon there being at least two or more requests for the resource.

Ali teaches a distributed file caching system that comprises multiple caches. Ali teaches that the caches can be turned on or turned off. In particular, Ali teaches that all of the caches can be turned ON or OFF – signified by a "global" flag — or that individual caches can be turned ON or OFF — signified by a "local" flag. Clearly, the fact that a cache can be turned ON or OFF isn't at all related to the recited limitation that — implicitly when the cache is ON — the cache is not populated with a resource until at least two requests for the resource have been received.

For this reason, the applicants respectfully submit that the rejection of claim 1 is traversed.

Because claims 2 through 7 depend on claim 1, the applicants respectfully submit that the rejection of them is also traversed.

Claim 8 recites:

> **8.** A data processing system comprising:
>
> a cache for storing a resource; and
>
> a processor for *populating said cache with said resource only when at least i requests for said resource have been received*;
>
> wherein *i* is an integer and is at least occasionally greater than one.
>
> (*emphasis supplied*)

For the same reasons as those given with respect to claim 1, nowhere does Ali teach or suggest, alone or in combination with the other references, what claim 8 recites – namely, populating a cache with a resource only upon the successful resolution of a test – the test (at least occasionally) being the reception of two or more requests for the resource.

Because claims 9 through 14 depend on claim 8, the applicants respectfully submit that the rejection of them is also traversed.

Claim 15 recites:

> **15.** A method comprising:
>
> receiving at a first node in a computer network at least one request for a resource;
>
> retrieving said resource from a second node in said computer network; and
>
> *populating a cache in said first node with said resource only when at least i requests for said resource have been received at said first node*;
>
> wherein *i* is an integer and is at least occasionally greater than one.
>
> (*emphasis supplied*)

For the same reasons as those given with respect to claim 1, nowhere does Ali teach or suggest, alone or in combination with the other references, what claim 15 recites – namely, populating a cache with a resource only upon the successful resolution of a test – the test (at least occasionally) being the reception of two or more requests for the resource.

Because claims 16 through 23 depend on claim 15, the applicants respectfully submit that the rejection of them is also traversed.

Claim 24 recites:

> **24.** A first node in a computer network, said first node comprising:
>
> a cache;
>
> at least one receiver for receiving at least one request for a resource; and
>
> a processor for retrieving said resource from a second node in said computer network, and f*or populating said cache in said first node with said resource only when at least i requests for said resource have been received at said first node*;
>
> wherein *i* is an integer and is at least occasionally greater than one.
>
> (*emphasis supplied*)

For the same reasons as those given with respect to claim 1, nowhere does Ali teach or suggest, alone or in combination with the other references, what claim 24 recites –

namely, populating a cache with a resource only upon the successful resolution of a test – the test (at least occasionally) being the reception of two or more requests for the resource.

Because claims 26 through 32 depend on claim 24, the applicants respectfully submit that the rejection of them is also traversed.

Ground 3: 35 U.S.C. 102 Rejection of Claims 1-32

Claims 1 through 32 have been rejected under 35 U.S.C. 102(e) as being anticipated by V. Tran et al., U.S. Patent 7,039,683 B1 (hereinafter "**Tran**").

Claim 1 recites:

> **1.** A method comprising:
>
> *populating a cache with a resource only when at least i requests for said resource have been received*;
>
> wherein *i* is an integer and is at least occasionally greater than one.
>
> (*emphasis supplied*)

Nowhere does Tran teach or suggest, alone or in combination with the other references, what claim 1 recites – namely, populating a cache with a resource only upon the successful resolution of a test – the test (at least occasionally) being the reception of two or more requests for the resource.

Tran teaches a variety of factors for determining when to pre-fetch a resource — in other words, when to put a resource into a cache before a request has arrived for the resource. The first sentence of Tran makes this perfectly clear:

> In one general aspect, electronic information caching may make electronic information more readily available to an access requestor based on an *anticipated demand* for that information.
>
> Tran Col. 1, lines 28-31 (*emphasis supplied*)

Pre-filling and "delayed" post-filling — as the present invention might be called — are completely different.

In general, pre-filling is difficult to do well, and Tran teaches a number of factors for deciding whether or not to pre-fetch a reference. These are described in the Summary of the Invention:

> *Anticipating future requests* for access to selected electronic information may be based on: *past requests for access to the same* or related electronic information by access requesters; past requests for access to non-related electronic information by access requestors; past requests for access to related non-electronic information by access requesters; or, past requests for access to non-related non-electronic information by access requesters. The relevant access activity that is measured may include the frequency or volume of requests for access to the selected electronic information.

> Anticipating requests for access also may determine the file size to assign a cache value based on the file size and the frequency of requests for the selected electronic information. Thus, selected electronic information with a higher cache value is more likely to be anticipated. Anticipating requests for access may also take into account criteria that are unrelated to past access requests. Requests for access may be anticipated before, after, or while an access request is made.
>
> Tran Col. 1, lines 49-56 (*emphasis supplied*)

As the first three words of the section point out, anticipating future requests may be based on past requests for the same resource.

The Office's argument to substantiate the rejection is based on a misreading of language in the detailed description. The Office action cites Col. 4, lines 25-42, which states:

> The *anticipating module* 1110 generally includes a segment of software code that is *designed to predict requests* for electronic information stored on an accessible source storage medium. For example, the software code may be designed to store requests for electronic information or to access a data storage device (located on the Web or otherwise) that stores requests for electronic information. *The software may be designed to measure the stored data to determine the rate or number of requests for the electronic information* and the file size of the electronic. *The software also may calculate a cache value based on the number of requests* and the file size, and to electronically flag the electronic information as being in high demand by access requesters where the cache value or frequency of requests for the electronic information is high. The anticipating module 1110 may predict requests for electronic information based on any or all of the criteria described with reference to step 1010 of FIG. 1.
>
> Tran Col. 4, lines 25-42 (*emphasis supplied*)

In other words, module 1110 predicts whether to pre-fetch a resource for a cache based on the number of previous requests for the cache. Again, this is not the same as the present invention for two reasons. First, the present invention requires at least one request for the resource, which Tran and pre-filling does not, and second, nowhere does Tran require that — under some circumstances — there must be at least two requests for the resource before it is put into the cache.

For this reason, the applicants respectfully submit that the rejection of claim 1 is traversed.

Because claims 2 through 7 depend on claim 1, the applicants respectfully submit that the rejection of them is also traversed.

Claim 8 recites:

> **8.** A data processing system comprising:
>
> a cache for storing a resource; and
>
> a processor for *populating said cache with said resource only when at least i requests for said resource have been received*;
>
> wherein *i* is an integer and is at least occasionally greater than one.
>
> (*emphasis supplied*)

For the same reasons as those given with respect to claim 1, nowhere does Tran teach or suggest, alone or in combination with the other references, what claim 8 recites – namely, populating a cache with a resource only upon the successful resolution of a test – the test (at least occasionally) being the reception of two or more requests for the resource.

Because claims 9 through 14 depend on claim 8, the applicants respectfully submit that the rejection of them is also traversed.

Claim 15 recites:

> **15.** A method comprising:
>
> receiving at a first node in a computer network at least one request for a resource;
>
> retrieving said resource from a second node in said computer network; and
>
> *populating a cache in said first node with said resource only when at least i requests for said resource have been received at said first node*;
>
> wherein *i* is an integer and is at least occasionally greater than one.
>
> (*emphasis supplied*)

For the same reasons as those given with respect to claim 1, nowhere does Tran teach or suggest, alone or in combination with the other references, what claim 15 recites – namely, populating a cache with a resource only upon the successful resolution of a test – the test (at least occasionally) being the reception of two or more requests for the resource.

Because claims 16 through 23 depend on claim 15, the applicants respectfully submit that the rejection of them is also traversed.

Claim 24 recites:

> **24.** A first node in a computer network, said first node comprising:
>
> a cache;
>
> at least one receiver for receiving at least one request for a resource; and
>
> a processor for retrieving said resource from a second node in said computer network, and f*or populating said cache in said first node with said resource only when at least i requests for said resource have been received at said first node*;
>
> wherein *i* is an integer and is at least occasionally greater than one.
>
> (*emphasis supplied*)

For the same reasons as those given with respect to claim 1, nowhere does Tran teach or suggest, alone or in combination with the other references, what claim 24 recites – namely, populating a cache with a resource only upon the successful resolution of a test – the test (at least occasionally) being the reception of two or more requests for the resource.

Because claims 26 through 32 depend on claim 24, the applicants respectfully submit that the rejection of them is also traversed.

CONCLUSION

The applicants have demonstrated that the logic underlying the Office's rejection is untenable, and, therefore, that the rejection is not sustainable. For this reason, the applicants respectfully request the Board of Appeals to reverse the decision of the Examiner as provided for in 37 C.F.R. 41.50(a).

Respectfully,
Peter Joseph Giacomini et al.

By **/Jason Paul DeMont/**
Reg. No. 35793
Attorney for Applicants
732-578-0103 x11

DeMont & Breyer, L.L.C.
Suite 250
100 Commons Way
Holmdel, NJ 07733
United States of America

Claims Appendix

1. (Previously Presented) A method comprising:

populating a cache with a resource only when at least *i* requests for said resource have been received;

wherein *i* is an integer and is at least occasionally greater than one.

2. (Original) The method of claim 1 wherein the value of *i* is invariant.

3. (Original) The method of claim 1 wherein the value of *i* is based on calendrical time.

4. (Original) The method of claim 1 wherein said cache is populated with said resource only when at least *i* requests for said resource have been received within an elapsed time interval, Δt.

5. (Original) The method of claim 4 wherein the duration of said elapsed time interval, Δt, is based on the value of *i*.

6. (Original) The method of claim 4 wherein the value of *i* is based on calendrical time.

7. (Original) The method of claim 4 wherein the duration of said elapsed time interval, Δt, is based on calendrical time.

---

8. (Previously Presented) A data processing system comprising:

a cache for storing a resource; and

a processor for populating said cache with said resource only when at least *i* requests for said resource have been received;

wherein *i* is an integer and is at least occasionally greater than one.

9. (Original) The data processing system of claim 8 wherein the value of *i* is invariant.

10. (Original) The data processing system of claim 8 wherein the value of *i* is based on calendrical time.

11. (Original) The data processing system of claim 8 wherein said cache is populated with said resource only when at least *i* requests for said resource have been received within an elapsed time interval, Δt.

12. (Original) The data processing system of claim 8 wherein the duration of said elapsed time interval, Δt, is based on the value of *i*.

13. (Original) The data processing system of claim 8 wherein the value of *i* is based on calendrical time.

14. (Original) The data processing system of claim 8 wherein the duration of said elapsed time interval, Δt, is based on calendrical time.

---

15. (Previously Presented) A method comprising:

receiving at a first node in a computer network at least one request for a resource;

retrieving said resource from a second node in said computer network; and

populating a cache in said first node with said resource only when at least *i* requests for said resource have been received at said first node;

wherein *i* is an integer and is at least occasionally greater than one.

16. (Original) The method of claim 15 wherein the value of *i* is invariant.

17. (Original) The method of claim 15 wherein the value of *i* is based on calendrical time.

18. (Original) The method of claim 15 wherein said cache is populated with said resource only when at least *i* requests for said resource have been received within an elapsed time interval, Δt.

19. (Original) The method of claim 18 wherein the duration of said elapsed time interval, Δt, is based on the value of *i*.

20. (Original) The method of claim 18 wherein the value of *i* is based on calendrical time.

21. (Original) The method of claim 18 wherein the duration of said elapsed time interval, Δt, is based on calendrical time.

22. (Original) The method of claim 15:

wherein said computer network is a hierarchical computer network and said first node has *m* filial nodes;

wherein said cache is populated with said resource only when at least one request for said resource has been received from at least *n* of said *m* filial nodes; and

wherein *m* is an integer greater than one, *n* is an integer greater than one, and $m \geq n$.

23. (Original) The method of claim 15:

wherein said computer network is a hierarchical computer network and said first node has *m* filial nodes;

wherein said cache is populated with said resource only when at least one request for said resource has been received from at least *n* of said *m* filial nodes within an elapsed time interval, Δt; and

wherein *m* is an integer greater than one, *n* is an integer greater than one, and $m \geq n$.

---

24. (Previously Presented) A first node in a computer network, said first node comprising:

a cache;

at least one receiver for receiving at least one request for a resource; and

a processor for retrieving said resource from a second node in said computer network, and for populating said cache in said first node with said resource only when at least *i* requests for said resource have been received at said first node;

wherein *i* is an integer and is at least occasionally greater than one.

25. (Original) The first node of claim 24 wherein the value of *i* is invariant.

26. (Original) The first node of claim 24 wherein the value of *i* is based on calendrical time.

27. (Original) The first node of claim 24 wherein said cache is populated with said resource only when at least *i* requests for said resource have been received within an elapsed time interval, Δt.

28. (Original) The first node of claim 27 wherein the duration of said elapsed time interval, Δt, is based on the value of *i*.

29. (Original) The first node of claim 27 wherein the value of *i* is based on calendrical time.

30. (Original) The first node of claim 27 wherein the duration of said elapsed time interval, Δt, is based on calendrical time.

31. (Original) The first node of claim 24:

wherein said computer network is a hierarchical computer network and said first node has *m* filial nodes;

wherein said cache is populated with said resource only when at least one request for said resource has been received from at least *n* of said *m* filial nodes; and

wherein *m* is an integer greater than one, *n* is an integer greater than one, and $m \geq n$.

32. (Original) The first node of claim 24:

wherein said computer network is a hierarchical computer network and said first node has *m* filial nodes;

wherein said cache is populated with said resource only when at least one request for said resource has been received from at least *n* of said *m* filial nodes within an elapsed time interval, Δt; and

wherein *m* is an integer greater than one, *n* is an integer greater than one, and $m \geq n$.

Evidence Appendix

There is no evidence submitted pursuant to 37 CFR §§ 1.130, 1.131, or1.132.

Related Proceedings Appendix

There are no related proceedings.